intercepting API function calls issued by said software component by replacing the addresses of API functions to be intercepted in an import data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different API function;

intercepting non-API function calls issued by said software component by replacing the addresses of non-API functions to be intercepted in an import data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different non-API function;

creating a call chain operative to permit distinguishing between function calls made by said software component from function calls made by said monitored application;

blocking intercepted API calls that are forbidden according to the security policy; and

allowing intercepted API calls that are permitted according to the security policy.

A1 Cont.

2. (Amended) The method according to claim 1, [wherein said step of intercepting comprises the steps of:] further comprising the step of injecting a security monitor implementing said secure sandbox into the address space of the monitored application[; and

redirecting said preselected set of API calls issued by the software component to said security monitor].

3. (Amended) The method according to claim 1, wherein said step of blocking intercepted API calls comprises the step of [blocking intercepted] preventing the execution of API calls that are in [the preselected] said selected set of [APIs] API function calls which have been determined to have originated by said software component and which have been determined to be forbidden according to the security policy.

4. (Amended) The method according to claim 1, wherein said step of allowing intercepted API calls comprises the step of allowing intercepted API calls that [are in the preselected set of APIs] have been determined to have originated by said monitored application and which are permitted according to the security policy.

33

6 ~~5~~. (Amended) A method of monitoring the execution of [a] an application and one or more software [component] components associated [with an application] therewith in accordance with a predetermined security policy, said method comprising the steps of:

[intercepting a preselected set of application programming interface (API) calls issued by the application;

intercepting non-API calls issued by the software component;]

intercepting a selected set of application programming interface (API) function calls issued by said monitored application by replacing the addresses of all API functions to be intercepted in an import data table associated with said monitored application with addresses of security monitor functions, each security monitor function associated with a different API function;

intercepting API function calls issued by said software component by replacing the addresses of API functions to be intercepted in an import data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different API function;

intercepting non-API function calls issued by said software component by replacing the addresses of non-API functions to be intercepted in an import data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different non-API function;

C1 Cont.

determining whether an intercepted API call issued by [the] said monitored application originated from a non-API call issued by the software component via the generation of a call chain by said software component when a non-API function is called;

blocking intercepted API calls that originated with a non-API call from the software component that are forbidden according to the security policy; and

allowing intercepted API calls that originated with a non-API call from the software component that are permitted according to the security policy.

7 ~~6~~. (Amended) The method according to claim 6 ~~5~~, [wherein said step of intercepting a preselected set of API calls issued by the application comprises the steps of:] further comprising the step of injecting a security monitor into the address space of the monitored application[; and

redirecting said preselected set of API calls issued by the application to said security monitor].

8 ~~7~~. (Amended) The method according to claim 6 ~~5~~, [wherein said step of intercepting non-API calls issued by the software component comprises the steps of:] further comprising the step of injecting a security monitor into the address space of said monitored application, said step of injecting comprising the steps of:

launching said monitored application in suspend mode;

allocating memory in the address space of said monitored application;

copying a loading function to said allocated memory;

creating a thread operative to execute said loading function which in turn functions to load said security monitor;

installing means for the interception of application programming interface (API) function calls made by said monitored application and said software component, and non-API function calls made by said software component;

unsuspending said thread and deallocating memory;

unsuspending said monitored application and permitting it to execute.

[injecting a security monitor into the address space of the monitored application; and

redirecting said non-API calls issued by the software component to said security monitor.]

C1 Cont.

10 ~~8~~. (Amended) A method of monitoring the execution of [a] an application and one or more software [component] components associated [with an application] therewith in accordance with a predetermined security policy, said method comprising the steps of:

injecting a security monitor into the address space of [the] said monitored application;

generating a plurality of stub functions corresponding to application programming interface (API) function calls and non-API [functions] function calls which are called by the software component;

redirecting all API calls and all non-API calls made by the software component;

redirecting API calls made by [the] said monitored application to said security monitor;

B

— setting a flag when [with each call made by the] said software component makes a call to either an API function or a non-API function;

redirecting a portion of API calls received by said plurality of stub functions to said security monitor;

redirecting said non-API calls made by the software component to their corresponding non-API functions; and

applying the predetermined security policy to an API call when said flag is set.

12 ~~9~~. (Amended) A method of monitoring the execution of [a] an application and one or more software [component] components associated [with an application] therewith in accordance with a predetermined security policy, said method comprising the steps of:

applying interception to the application including all its modules whether loaded initially or during execution thereof;

detecting the loading of a software component external to the application;

applying interception to all calls made by the software component to functions located in other modules; and

applying the security policy to said calls made by the software component.

C1 Cont'd.

14 ~~10~~. (Amended) A method of monitoring the execution of [a] an application and one or more software [component] components associated [with an application] therewith in accordance with a predetermined security policy, said method comprising the steps of:

[applying] installing means for interception [to the] within said monitored application including all [its] modules associated therewith whether loaded initially or during execution thereof;

detecting the loading of a software component external to [the] said monitored application;

[applying interception] installing means for intercepting [to] all API and non-API function calls made by the software component to functions located in other modules; ~~and~~

B

setting a flag when a function call is issued by the software component to any function located in another module located external thereto; and

[applying interception to API calls contained in a preselected set; and]

applying the security policy to an API call when said flag is set.

36

L

Please add new claims ~~11~~ to ~~19~~.

5 ~~11~~. (New) The method according to claim 1, wherein said software component comprises one of the following: ActiveX control, Java component and Netscape Plugin component.

9 ~~12~~. (New) The method according to claim 6 ~~5~~, wherein said software component comprises one of the following: ActiveX control, Java component and Netscape Plugin component.

11 ~~13~~. (New) The method according to claim 10 ~~8~~, wherein said software component comprises one of the following: ActiveX control, Java component and Netscape Plugin component.

13 ~~14~~. (New) The method according to claim 12 ~~9~~, wherein said software component comprises one of the following: ActiveX control, Java component and Netscape Plugin component.

15. (New) The method according to claim 14 ~~10~~, wherein said software component comprises one of the following: ActiveX control, Java component and Netscape Plugin component.

17 ~~16~~. (New) A method of creating a secure sandbox around both a monitored application and one or more software components associated therewith in accordance with a predetermined security policy, said method comprising the steps of:

a2 Cont.

intercepting a selected set of application programming interface (API) function calls issued by said monitored application by replacing the addresses of all API functions to be intercepted in an import data table associated with said monitored application with addresses of security monitor functions, each security monitor function associated with a different API function;

detecting a load type API function call issued by said monitored application;

blocking intercepted API calls that are forbidden according to the security policy; and

allowing intercepted API calls that are permitted according to the security policy.

18 ~~17~~. (New) The method according to claim 17 ~~16~~, wherein said load type API function call comprises one API function call from the group consisting of CoGetClassObject(), LoadLibrary() and LoadLibraryEx().

19 ~~18~~. (New) The method according to claim 17 ~~16~~, further comprising the steps of:

upon detection of a load type API function call:

intercepting API function calls issued by said software component by replacing the addresses of API functions to be intercepted in an import

37

data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different API function;

intercepting non-API function calls issued by said software component by replacing the addresses of non-API functions to be intercepted in an import data table associated with said software component with addresses of stub functions, each stub function operative to call a security monitor function associated with a different non-API function; and

a2 Cold.

creating a call chain operative to permit distinguishing between function calls made by said software component from function calls made by said monitored application.

~~19~~16. (New) The method according to claim ~~10~~14, wherein said step of detecting comprises the step of detecting one of the API functions from the group consisting of CoGetClassObject(), LoadLibrary() and LoadLibraryEx().